rfishbase: exploring, manipulating and visualizing FishBase data from R

Carl Boettiger[a,*], Duncan Temple Lang[b], Peter C. Wainwright[a]

[a] *Center for Population Biology, University of California, Davis, California 95616*

[b] *Department of Statistics, University of California, Davis, California 95616*

Abstract

This paper introduces a package that provides interactive and programmatic access to the FishBase repository. This package allows one to interact with data on over 30,000 fish species in the rich statistical computing environment, `R`. This direct, scriptable interface to FishBase data enables better discovery and integration essential for large-scale comparative analyses. The paper provides several examples to illustrate how the package works, and how it can be integrated into phylogenetics packages such as `ape` and `geiger`.

keywords R | vignette | fishbase

Introduction

FishBase (fishbase.org) is an award-winning online database of information about the morphology, trophic ecology, physiology, ecotoxicology, reproduction and economic relevance of the world's fishes, organized by species (Froese and Pauly 2012). This repository of information has proven to be a profoundly valuable community resource and the data has the potential to be used in a wide range of studies. However, assembling subsets of data housed in FishBase for use in focused analyses can be tedious and time-consuming. To facilitate the extraction, visualization, and integration of this data, the `rfishbase` package was been written for the R language for statistical computing and graphics (R Development Core Team 2012). R is a freely available open source computing environment that is used extensively in ecological research, with a large collection of packages built explicitly for this purpose (Kneib 2007).

The `rfishbase` package is dynamically updated from the FishBase database, describe its functions for extracting, manipulating and visualizing data, and then illustrate how these functions can be combined for more complicated analyses. Lastly it illustrates how having access to FishBase data through R allows a user to interface with other resources such as comparative phylogenetics software. The purpose of this paper is to introduce `rfishbase` and illustrate core features of its functionality.

Accessing FishBase data from R

In addition to its web-based interface, FishBase provides machine readable XML files for 30,622 (as accessed on 14 May, 2012) of its species entries to facilitate programmatic access of the data housed in this resource. As complete downloads of the FishBase database are not available, the FishBase team encouraged us to use these XML files as an entry point for programmatic access. We have introduced caching and pausing features into the package to prevent access from overtaxing the FishBase servers.
While FishBase encourages the use of its data by programmatic access, (Froese, pers. comm), users of `rfishbase` should

[*] Corresponding author, cboettig@ucdavis.edu
Email address: `cboettig@ucdavis.edu` (Peter C. Wainwright)

respect these load-limiting functions, and provide appropriate acknowledgment. A more detailed discussion of incentives, ethics, and legal requirements in sharing and accessing such repositories can be found in the respective literature, *e.g.* Fisher and Fortmann (2010) or Costello (2009).

The `rfishbase` package works by creating a cached copy of all data on FishBase currently available in XML format on the FishBase webpages. This process relies on the RCurl (Lang 2012a) and XML (Lang 2012b) packages to access these pages and parse the resulting XML into a local cache. Caching increases the speed of queries and greatly reduces demands on the FishBase server, which in its present form is not built to support direct access to application programming interfaces (APIs). A cached copy is included in the package and can be loaded in to R using the command:

```
data(fishbase)
```

This loads a copy of all available data from FishBase into the R list, `fish.data`, which can be passed to the various functions of `rfishbase` for extraction, manipulation and visualization. The online repository is frequently updated as new information is uploaded. To get the most recent copy of FishBase, update the cache instead. The update may take up to 24 hours. This copy is stored in the specified directory (note that "." can be used to indicate the current working directory) with the current date. The most recent copy of the data in the specified path can be loaded with the `loadCache()` function. If no cached set is found, `rfishbase` will load the copy originally included in the package.

```
updateCache(".")
loadCache(".")
```

Loading the database creates an object called fish.data, with one entry per fish species for which data was successfully found, for a total of 30,622 species.

Not all the data available in FishBase is included in these machine-readable XML files. Consequently, `rfishbase` returns taxonomic information, trophic description, habitat, distribution, size, life-cycle, morphology and diagnostic information. The information returned in each category is provided as plain-text, consequently `rfishbase` must use regular expression matching to identify the occurrence of particular words or patterns in this text corresponding to data of interest (Friedl 2006). Any regular expression can be used in in search queries. While these expressions allows for very precise pattern matching, applying this approach to plain text runs some risk of error which should not be ignored. Visual inspection of matches and careful construction of these expressions can help mitigate this risk. We provide example functions for reliably matching several quantitative traits from these text-based descriptions, which can be used as a basis for writing functions to identify other terms of interest.

Quantitative traits such as standard length, maximum known age, spine and ray counts, and depth information are provided consistently for most species, allowing `rfishbase` to extract this data directly. Other queries require pattern matching. While simple text searches within a given field are usually reliable, the `rfishbase` search functions will take any regular expression query, which permits logical matching, identification of number strings, and much more. The interested user should consult a reference on regular expressions after studying the simple examples provided here to learn more.

Tools for data extraction, analysis, and visualization

The basic tool for data extraction in `rfishbase` is the `which_fish()` function. This function takes a list of FishBase data (usually the entire database, `fish.data`, or a subset thereof, as illustrated later) and returns an array of those species matching the query. This array is given as a list of true/false values for every species in the query. This return structure has several advantages which are illustrated below.

Here is a query for reef-associated fish (mention of "reef" in the habitat description), and second query for fish that have "nocturnal" in their trophic description:

```
reef <- which_fish("reef", "habitat", fish.data)
nocturnal <- which_fish("nocturnal", "trophic", fish.data)
```

One way these returned values are commonly used is to obtain a subset of the database that meets this criteria, which can then be passed on to other functions. For instance, if one wants the scientific names of these reef fish, one can use the `fish_names` function. Like the `which_fish` function, it takes the list of FishBase data, `fish.data` as input. In this example, just the subset that are reef affiliated are passed to the function,

```
reef_species <- fish_names(fish.data[reef])
```

Because our `reef` object is a list of logical values (true/false), one can combine this in intuitive ways with other queries. For instance, one can query for the names of all fish that are both nocturnal and not reef associated,

```
nocturnal_nonreef_orders <- fish_names(fish.data[nocturnal & !reef], ”Class”)
```

Note that in this example, it is also specified that the user wants the taxonomic Class of the fish matching the query, rather than the species names. `fish_names` will allow the user to specify any taxonomic level for it to return. Quantitative trait queries work in a similar manner to `fish_names`, taking the FishBase data and returning the requested information. For instance, the function `getSize` returns the length (default), weight, or age of the fish in the query:

```
age <- getSize(fish.data, ”age”)
```

`rfishbase` can also extract a table of quantitative traits from the morphology field, describing the number of vertebrate, dorsal and anal fin spines and rays,

```
morphology_numbers <- getQuantTraits(fish.data)
```

and extract the depth range (extremes and usual range) from the habitat field,

```
depths <- getDepth(fish.data)
```

A list of all the functions provided by `rfishbase` can be found in Table 1.
The `rfishbase` manual provided with the package provides more detail about each of these functions, together with examples for their use.

The real power of programmatic access is the ease with which one can combine, visualize, and statistically test a custom compilation of this data. To do so it is useful to organize a collection of queries into a data frame. The next set of commands combines the queries made above and a few additional queries into a data frame in which each row represents a species and each column represents a variable.

```
marine <- which_fish(”marine”, ”habitat”, fish.data)
africa <- which_fish(”Africa:”, ”distribution”, fish.data)
length <- getSize(fish.data, ”length”)
order <- fish_names(fish.data, ”Order”)
dat <- data.frame(reef, nocturnal,  age, marine, africa, length, order)
```

This data frame contains categorical data (*e.g.* is the fish a carnivore) and continuous data (*e.g.* weight or age of fish). One can take advantage of data visualization tools in R to begin exploring this data. These examples are simply meant to be illustrative of the kinds of analysis possible and how they would be constructed.

For instance, one can identify which orders contain the greatest number of species, and for each of them, plot the fraction in which the species are marine.

```
biggest <- names(head(sort(table(order),decr=T), 8))
primary_orders <- subset(dat, order %in% biggest)
```

function.name	description
familySearch	A function to find all fish that are members of a scientific Family
findSpecies	Returns the matching indices in the data given a list of species names
fish.data	A cached copy of extracted FishBase data, 03/2012.
fish_names	Return the scientific names, families, classes, or orders of the input data
getDepth	Returns available depth range data
getQuantTraits	Returns all quantitative trait values found in the morphology data
getRefs	Returns the FishBase reference id numbers matching a query.
getSize	Returns available size data of specified type (length, weight, or age)
habitatSearch	A function to search for the occurances of any keyword in habitat description
labridtree	An example phylogeny of labrid fish
loadCache	Load an updated cache
updateCache	Update the cached copy of fishbase data
which_fish	which_fish is the the generic search function for fishbase a variety of description types

Table 1: A list of each of the functions and data objects provided by rfishbase

```
ggplot(primary_orders, aes(order, fill=marine)) + geom_bar() +
# a few commands to customize appearance
  geom_bar(colour="black",show_guide=FALSE) +
  opts(axis.text.x=theme_text(angle=90, hjust=1, size=6)) +
  opts(legend.title=theme_blank(), legend.justification=c(1,0), legend.position=c(.9,.6)) +
  scale_fill_grey(labels=c("Marine", "Non-marine")) +
  xlab("") + ylab("Number of species")
```

FishBase data excels for comparative studies across many species, but searching through over 30,000 species to extract data makes broad comparative analyses quite time-consuming. Having access to the data in R, one can answer such questions as fast as they are posed. Consider looking for a correlation between the maximum age and the size of fish. One can partition the data by any variable of interest as well – this example color codes the points based on whether or not the species is marine-associated. The ggplot2 package (Wickham 2009) provides a particularly powerful and flexible language for visual exploration of such patterns.

```
ggplot(dat,aes(age, length, shape=marine)) +
  geom_point(position='jitter', size=1) +
  scale_y_log10() + scale_x_log10(breaks=c(50,100,200)) +
  scale_shape_manual(values=c(1,19), labels=c("Marine", "Non-marine")) +
  ylab("Standard length (cm)") + xlab("Maximum observed age (years)") +
  opts(legend.title=theme_blank(), legend.justification=c(1,0), legend.position=c(.9,0)) +
  opts(legend.key = theme_blank())
```

A wide array of visual displays are available for different kinds of data. A box-plot is a natural way to compare the distributions of categorical variables, such as asking “Are reef species longer lived than non-reef species in the marine environment?”

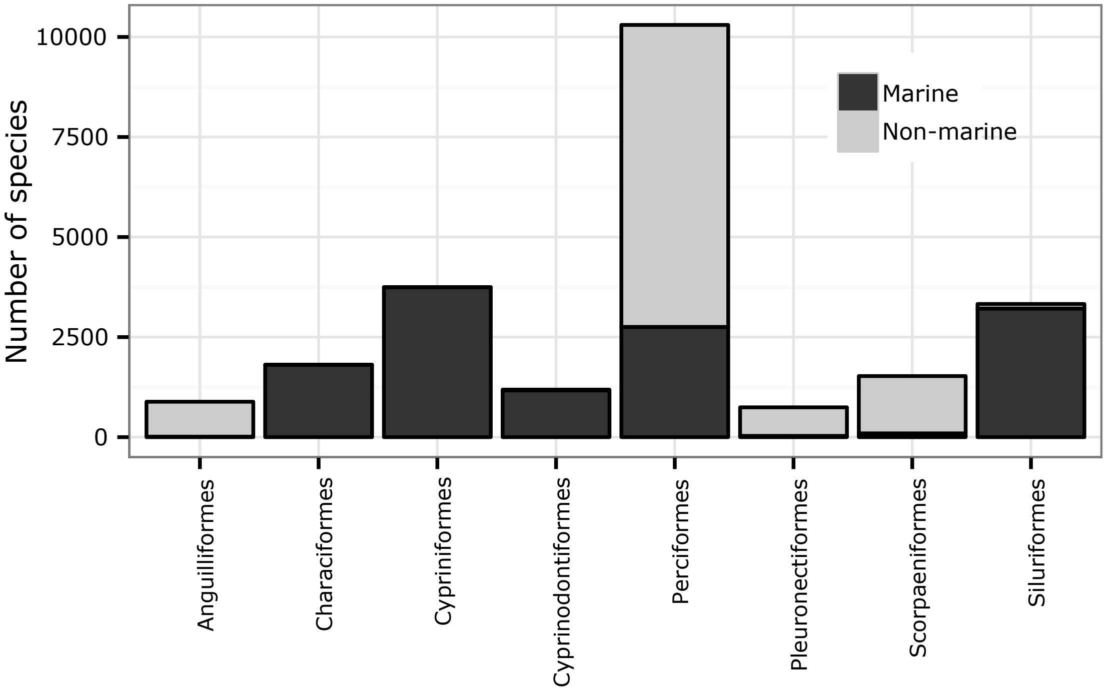

Figure 1: Fraction of marine species in the eight largest orders of teleost fishes

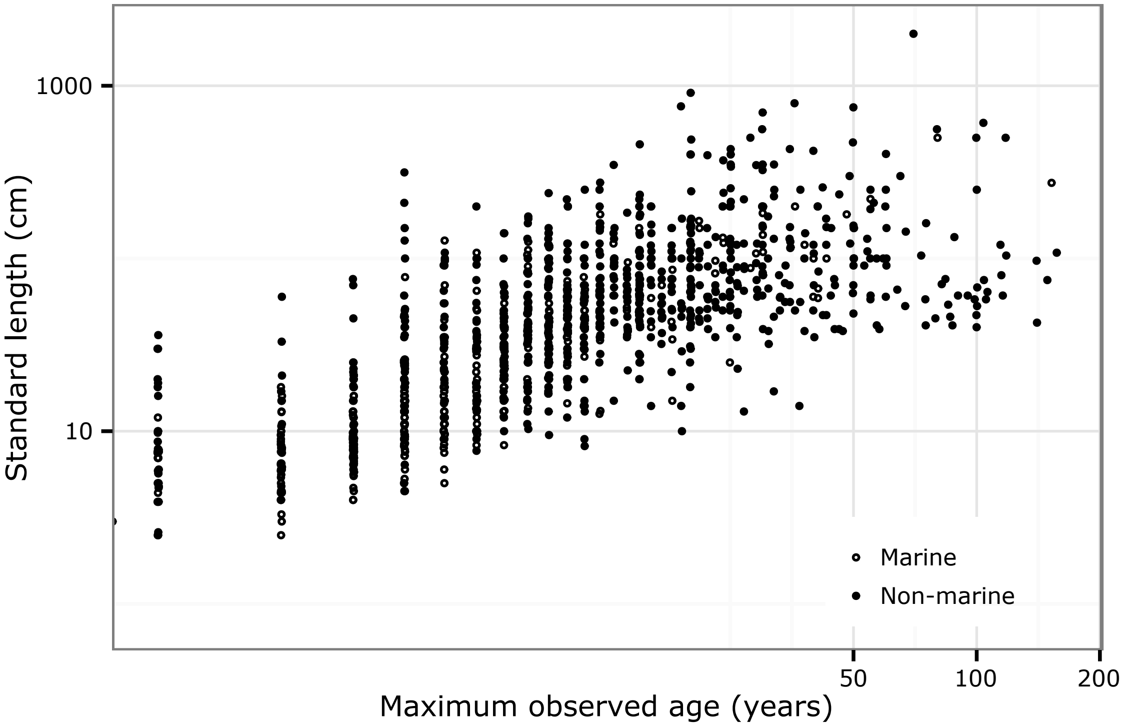

Figure 2: Scatterplot maximum age with length observed in each species. Color indicates marine or freshwater species.

```
ggplot(subset(dat, marine)) +
  geom_boxplot(aes(reef, age)) +
  scale_y_log10() + xlab("") +
  ylab("Maximum observed age (years)")  +
  opts(axis.text.x = theme_text(size = 8))
```

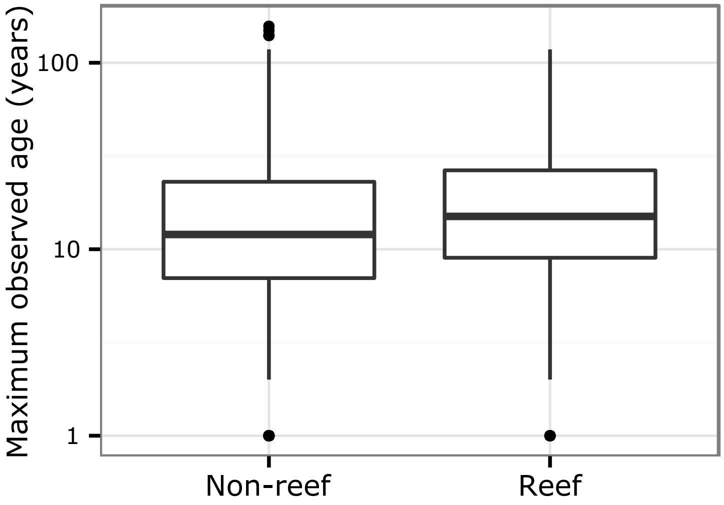

Figure 3: Distribution of maximum age for reef-associated and non-reef associated fish

In addition to powerful visualizations R provides an unparalleled array of statistical analysis methods. Executing the linear model testing the correlation of length with maximum size takes a single line,

```
library(MASS)
corr.model <- summary(rlm(data=dat,  length ~ age))
```

which shows a significant correlation between maximum age and standard length (P = 0.001).

Comparative studies

Many ecological and evolutionary studies rely on comparisons between taxa to pursue questions that cannot be approached experimentally. For instance, recent studies have attempted to identify whether reef-associated clades experience greater species diversification rates than non-reef-associated groups (*e.g.* Alfaro et al. 2009). One can identify and compare the numbers of reef associated species in different families using the `rfishbase` functions presented above.

In this example, consider the simpler question "Are there more reef-associated species in *Labridae* than in *Gobiidae*?" Recent research has shown that the families Scaridae and Odacidae are nested within Labridae (Westneat and Alfaro 2005), although the three groups are listed as separate families in fishbase. We get all the species in fishbase from *Labridae* (wrasses), *Scaridae* (parrotfishes) and *Odacidae* (weed-whitings):

```
labrid <- which_fish("(Labridae|Scaridae|Odacidae)", "Family", fish.data)
```

and get all the species of gobies

```
goby <- which_fish("Gobiidae", "Family", fish.data)
```

Identify how many labrids are found on reefs

```
labrid.reef <- which_fish(”reef”, ”habitat”, fish.data[labrid])
labrids.on.reefs <- table(labrid.reef)
```

and how many gobies are found on reefs:

```
gobies.on.reefs <- table(which_fish(”reef”, ”habitat”, fish.data[goby]) )
```

Note that summing the list of true/false values returned gives the total number of matches. This reveals that there are 505 labrid species associated with reefs, and 401 goby species associated with reefs. This example illustrates the power of accessing the FishBase data: Gobies are routinely listed as the biggest group of reef fishes (*e.g.* Bellwood and Wainwright 2002) but this is because there are more species in *Gobiidae* than any other family of reef fish. When one counts the species in each group that live on reefs one finds that labrids are actually the most species-rich family on reefs.

Integration of analyses

One of the greatest advantages of accessing FishBase directly through R is the ability to take advantage of other specialized analyses available through R packages. Users familiar with these packages can more easily take advantage of the data available on FishBase. This is illustrated with an example that combines phylogenetic methods available in R with quantitative trait data available from `rfishbase`.

This series of commands illustrates testing for a phylogenetically corrected correlation between the observed length of a species and the maximum observed depth at which it is found. One begins by reading in the data for a phylogenetic tree of labrid fish (provided in the package), and the phylogenetics packages `ape` (Paradis, Claude, and Strimmer 2004) and `geiger` (Harmon et al. 2009).

```
data(labridtree)
library(ape)
library(geiger)
```

Find the species represented on this tree in FishBase

```
myfish <- findSpecies(labridtree$tip.label, fish.data)
```

Get the maximum depth of each species and sizes of each species:

```
depths <- getDepth(fish.data[myfish])[,”deep”]
size <- getSize(fish.data[myfish], ”length”)
```

Drop missing data, and then drop tips from the phylogeny for which data was not available:

```
data <- na.omit(data.frame(size,depths))
pruned <- treedata(labridtree, data)
```

```
Dropped tips from the tree because there were no matching names in the data:
 [1] ”Anampses_geographicus”     ”Bodianus_perditio”
 [3] ”Chlorurus_bleekeri”        ”Choerodon_cephalotes”
 [5] ”Choerodon_venustus”        ”Coris_batuensis”
 [7] ”Diproctacanthus_xanthurus” ”Halichoeres_melanurus”
 [9] ”Halichoeres_miniatus”      ”Halichoeres_nigrescens”
[11] ”Macropharyngodon_choati”   ”Oxycheilinus_digrammus”
[13] ”Scarus_flavipectoralis”    ”Scarus_rivulatus”
```

Use phylogenetically independent contrasts (Felsenstein 1985) to determine if depth correlates with size after correcting for phylogeny:

```
corr.size <- pic(pruned$data[["size"]],pruned$phy)
corr.depth <- pic(pruned$data[["depths"]],pruned$phy)
corr.summary <- summary(lm(corr.depth ~ corr.size - 1))
```

which returns a non-significant correlation (p = 0.47).

```
ggplot(data.frame(corr.size,corr.depth), aes(corr.size,corr.depth)) +
 geom_point() + stat_smooth(method=lm, col=1) +
 xlab("Contrast of standard length (cm)") +
 ylab("Contrast maximum depth (m)") + opts(title="Phylogenetically standardized contrasts")
```

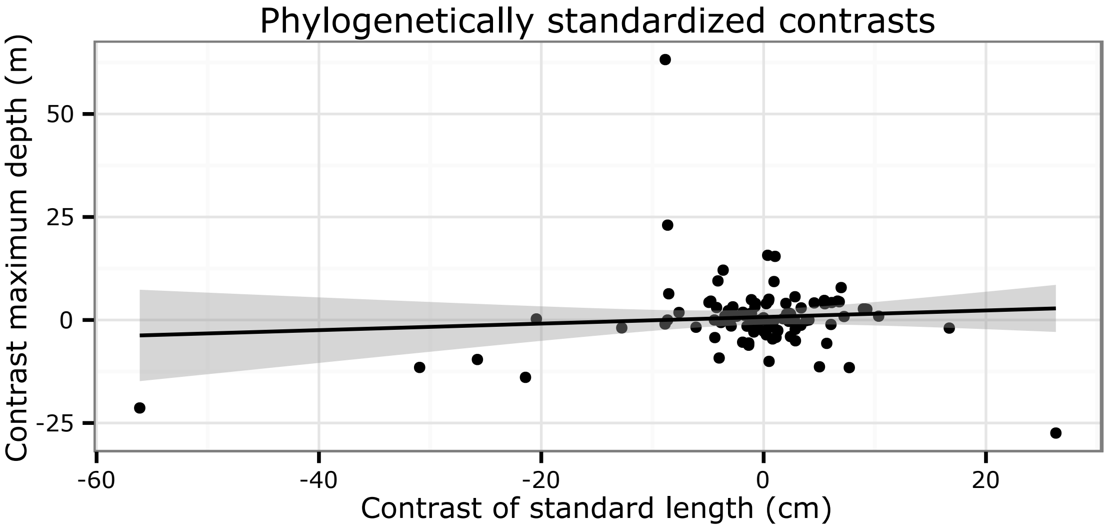

Figure 4: Correcting for phylogeny, size is not correlated with maximum depth observed in labrids

One can also estimate different evolutionary models for these traits to decide which best describes the data,

```
bm <- fitContinuous(pruned$phy, pruned$data[["depths"]], model="BM")[[1]]
ou <- fitContinuous(pruned$phy, pruned$data[["depths"]], model="OU")[[1]]
```

where the Brownian motion model has an AIC score of 1,185 while the OU model has a score of 918.2, suggesting that OU is the better model.

Discussion

With more and more data readily available, informatics is becoming increasingly important in ecology and evolution research (Jones et al. 2006), bringing new opportunities for research (Parr et al. 2011; Michener and Jones 2012) while also raising new challenges (Reichman, Jones, and Schildhauer 2011). It is in this spirit that the `rfishbase` package provides programmatic access to the data available on the already widely recognized database, FishBase. Such tools allow researchers to take greater advantage of the data available, facilitating deeper and richer analyses than would be feasible under only manual access to the data. The examples in this manuscript are intended to illustrate how this package works, and to help inspire readers to consider and explore questions that would otherwise be too time consuming or challenging to pursue. This paper has introduced the functions of the `rfishbase` package and described how they can be used to improve the extraction, visualization, and integration of FishBase data in ecological and evolutionary research.

The self-updating study

Because analyses using this data are written in R scripts, it becomes easy to update the results as more data becomes available on FishBase. Programmatic access to data coupled with script-able analyses can help ensure that research is more easily reproduced and also facilitate extending the work in future studies (Peng 2011; Merali 2010). This document is an example of this, using a dynamic documentation interpreter program which runs the code displayed to produce the results shown, decreasing the possibility for faulty code (Xie 2012). As FishBase is updated, one can regenerate these results with less missing data. Readers can find the original document which combines the source-code and text on the project's Github page.

Limitations and future directions

FishBase contains much data that has not been made accessible in machine-readable XML format. The authors are in contact with the database managers and look forward to providing access to additional types of data as they become available. Because most of the data provided in the XML comes as plain text rather that being identified with machine-readable tags, reliability of the results is limited by text matching.
Improved text matching queries could provide more reliable information, and facilitate other specialized queries such as extracting geographic distribution details as categorical variables or latitude/longitude coordinates. FishBase taxonomy is inconsistent with taxonomy provided elsewhere, and additional package functions could help resolve these differences in assignments.

`rfishbase` has been available to R users through the Comprehensive R Archive Network since October 2011, and has a growing user base. The project remains in active development to evolve with the needs of its users. Users can view the most recent changes and file issues with the package on its development website on Github, (https://github.com/ropensci/rfishbase) and developers can submit changes to the code or adapt it into their own software.

Programmers of other R software packages can make use of the `rfishbase` package to make this data available to their functions, further increasing the use and impact of FishBase. For instance, the project OpenFisheries makes use of the `rfishbase` package to provide information about commercially relevant species.

Acknowledgements

This work was supported by a Computational Sciences Graduate Fellowship from the Department of Energy under grant number DE-FG02-97ER25308 to CB and National Science Foundation grant DEB-1061981 to PCW. The `rfishbase` package is part of the rOpenSci project (ropensci.org).

References

Alfaro, Michael E., Francesco Santini, Chad D. Brock, Hugo Alamillo, Alex Dornburg, Daniel L. Rabosky, Giorgio Carnevale, and Luke J. Harmon. 2009. "Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates." *Proceedings of the National Academy of Sciences* 106 (aug): 13410–14. doi:10.1073/pnas.0811087106. http://www.pubmedcentral.nih.gov/artic

Bellwood, D. R., and Peter C. Wainwright. 2002. "The history and biogeography of fishes on coral reefs." In *Coral Reef Fishes. Dynamics and diversity in a complex ecosystem*, ed. P. F. Sale, 5–32. San Diego: Academic Press.

Costello, Mark J. 2009. "Motivating Online Publication of Data." *BioScience* 59 (may): 418–427. doi:10.1525/bio.2009.59.5.9. http://caliber.ucpress.net/doi/abs/10.1525/bio.2009.59.5.9 http://www.jstor.org/stable/25502450.

Felsenstein, Joseph. 1985. "Phylogenies and the Comparative Method." *The American Naturalist* 125 (jan): 1–15. doi:10.1086/284325. http://www.journals.uchicago.edu/doi/abs/10.1086/284325.

Fisher, Joshua B., and Louise Fortmann. 2010. "Governing the data commons: Policy, practice, and the advancement of science." *Information & Management* 47 (may): 237–245. doi:10.1016/j.im.2010.04.001. http://linkinghub.elsevier.com/retrieve/pii/S03787206

Friedl, Jeffrey E. F. 2006. *Mastering regular expressions*. O'Reilly Media, Inc.. http://books.google.com/books?id=NYEX-Q9evKoC\&pgis=1.

Froese, R., and Daniel Pauly. 2012. “FishBase.” World Wide Web electronic publication.. www.fishbase.org.

Harmon, Luke, Jason Weir, Chad Brock, Rich Glor, Wendell Challenger, and Gene Hunt. 2009. “geiger: Analysis of evolutionary diversification.” http://cran.r-project.org/package=geiger.

Jones, Matthew B., Mark P. Schildhauer, O. J. Reichman, and Shawn Bowers. 2006. “The New Bioinformatics: Integrating Ecological Data from the Gene to the Biosphere.” *Annual Review of Ecology, Evolution, and Systematics* 37 (dec): 519–544. doi:10.1146/annurev.ecolsys.37.091305.110031. http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.ecolsys.37.091305.110031

Kneib, Thomas. 2007. “Introduction to the Special Volume on ’Ecology and Ecological Modelling in R’.” *Journal of Statistical Software* 22: 1–7. http://www.jstatsoft.org/v22/i01/paper.

Lang, Duncan Temple. 2012a. “RCurl: General network (HTTP/FTP/...) client interface for R.” http://cran.r-project.org/package=RCurl.

———. 2012b. “XML: Tools for parsing and generating XML within R and S-Plus.” http://cran.r-project.org/package=XML.

Merali, Zeeya. 2010. “Why Scientific programming does not compute.” *Nature*: 6–8.

Michener, William K., and Matthew B. Jones. 2012. “Ecoinformatics: supporting ecology as a data-intensive science.” *Trends in Ecology & Evolution* 27 (jan): 85–93. doi:10.1016/j.tree.2011.11.016. http://linkinghub.elsevier.com/retrieve/pii/S016953471100339

Paradis, E., J. Claude, and K. Strimmer. 2004. “APE: analyses of phylogenetics and evolution in R language.” *Bioinformatics* 20: 289–290.

Parr, Cynthia S., Robert Guralnick, Nico Cellinese, and Roderic D. M. Page. 2011. “Evolutionary informatics: unifying knowledge about the diversity of life.” *Trends in ecology & evolution* 27 (dec): 94–103. doi:10.1016/j.tree.2011.11.001. http://www.ncbi.nlm.nih.gov/pubmed/22154516.

Peng, R. D. 2011. “Reproducible Research in Computational Science.” *Science* 334 (dec): 1226–1227. doi:10.1126/science.1213847. http://www.sciencemag.org/cgi/doi/10.1126/science.1213847.

R Development Core Team, The. 2012. “R: A language and environment for statistical computing.” Vienna, Austria: R Foundation for Statistical Computing. http://www.r-project.org/.

Reichman, O. J., Matthew B. Jones, and Mark P. Schildhauer. 2011. “Challenges and Opportunities of Open Data in Ecology.” *Science* 331 (feb): 703–705. doi:10.1126/science.1197962. http://www.sciencemag.org/cgi/doi/10.1126/science.1197962 http://www.ncbi.nlm.nih.gov/pubmed/21311007.

Westneat, Mark W., and Michael E. Alfaro. 2005. “Phylogenetic relationships and evolutionary history of the reef fish family Labridae.” *Molecular phylogenetics and evolution* 36 (aug): 370–90. doi:10.1016/j.ympev.2005.02.001. http://www.ncbi.nlm.nih.gov/pubm

Wickham, Hadley. 2009. *ggplot2: elegant graphics for data analysis*. Springer New York. http://had.co.nz/ggplot2/book.

Xie, Yihui. 2012. “knitr: A general-purpose package for dynamic report generation in R.” http://yihui.name/knitr/.